EECS 268 Programming II (labs)

Shriraj K. Vaidya

About Me

- Email : shriraj@ku.edu
- Office : EATN 2027
- Office Hours:
  - M 10:00 AM - 11:00 AM
  - W 2:00 PM - 3:00 PM

Lab Page

- https://wiki.ittc.ku.edu/ittc_wiki/index.php?title=EECS268:Labs

Overview

- Class Defining and Object Creation
- List of Objects
- File Manipulation
- Terminal I/O

Board Game Data Query

- Given an input file containing information about board games, provide an interface for querying information about the board games using OOP.
- Board game data is as following:
  <name> <gibbonsrating> <baverage> <avgweight> <yearpublished> <bggbestplayers>

Board Game Data Query

	name	gibbonsrating	baverage	avgweight	yearpublished	bggbestplayers
2	10 Minute Heist: The Wizard's Tower	6	5.83116	1.2308	2017	3
3	1st & Goal	7.5	5.96546	1.807	2011	2
4	7 Wonders Duel	8	7.97584	2.2264	2015	2
5	Animal Upon Animal	10	6.52574	1.0405	2005	4
6	Animal Upon Animal: Christmas Edition	8.5	5.61091	0	2020	2
7	Animal Upon Animal: Small and Yet Great!	7	5.64657	1	2011	2
8	Arkham Horror (Third Edition)	6.5	7.13902	3.3382	2018	3
9	Arkham Horror: The Card Game	9.5	7.91131	3.5262	2016	2
10	Arkham Horror: The Card Game – Return to the Night of the Zealot	3	6.17709	3.8	2018	2
11	Arkham Horror: The Card Game – The Dunwich Legacy: Expansion	7	7.34828	3.4528	2017	2
12	Arkham Horror: The Card Game – The Forgotten Age: Expansion	8	6.40284	3.8696	2018	2
13	Arkham Horror: The Card Game – The Path to Carcosa: Expansion	9	6.97267	3.52	2017	2
14	BANG! The Dice Game	7.5	6.75972	1.2806	2013	6,7
15	Bargain Quest	7	6.35358	2.0141	2017	4

QUERIES

1)Print all games highest Gibbons range to lowest
2)Print all games from a year
3)Print all games with a weight equal to or lower than provided weight
4)The People VS Dr. Gibbons
5)Print based on player count
6)Exit

PRINTING FORMAT

```
============
name
gibbonsrating
baverage
avgweight
yearpublished
bggbestplayers
============
```

```
Animal Upon Animal
10.0
6.52574
1.0405
2005
4
```

USEFUL INFORMATION

- Remember __str__() and __repr__(), this will help you to print the broad games that user query.
- with open(filename) - open a file from filename and returns the object.
- String.split(sep) – returns string split by the separator
- def __init__(self, …) – Defining the initializer method for a class
- class1 = MyClass() – Creating an instance from the class object.
- Add comments when necessary. (DOCSTRINGS)
- Think about your program structure and write clean code.
- Run and save your code often. (For easy debugging)

Files You'll Create

- driver.py -

```
from executive import Executive

def main():
  file_name = input("Enter the name of the input file: ")
  my_exec = Executive(file_name)
  my_exec.run()

if __name__ == "__main__":
  main()
```

Files You'll Create(2)

- executive.py
  - Executive class
  - Reading the file input (splitting data lines and tabs)
  - Creating Board Games
  - Printing Menu
  - In Charge of executing all the queries

Files You'll Create(3)

- boardgames.py
  - Blueprint for the boardgames
  - Initializing board game attributes
  - __str__ and __repr__ implementations for the board games

This lab is a refresher for EECS168.

RUBRICS

Grades will be assigned according to the following criteria:

- [10pts] Attendance
- [10pts] Class Design
  - Each class' role should be well defined
  - Avoid create "god classes" or classes that are in charge of way too much, break the problem down into several classes
- [5pts] Code documentation
  - Files need comments (e.g. Author, date, modified time)
  - Methods need descriptions as well
  - Readability
- [5pts] Readability of output and stability of interaction
  - Check for bad ranges on good types of input
- [50pts] User interactions
  - [10pts] interaction 1
  - [10pts] interaction 2
  - [10pts] interaction 3
  - [15pts] interaction 4
  - [13pts] interaction 5
  - [2pts] exiting
- [10pts] No random crashes (assume valid file names and format, but make sure the user isn't typing "catfood" when you expect a number)